Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listings of Claims:

Claims 1-66 (Cancelled).

67. (Currently Amended) A method of producing obfuscated object code, the method comprising: substituting an assignment of a variable in source code with a class template defining a plurality of functions of the variable, each of the plurality of functions indexed by a key value and associated with a series of operations resulting in the assignment of the variable in a manner obfuscating such assignment; ~~selected function of the variable,~~ and compiling the source code using a compiler to produce object code, wherein the compiler inserts in the object code the series of operations associated with one of the plurality of functions of the variable identified by a key value provided as a parameter of the class template. ~~selected function is arranged to cause the variable to be presented in the compiled object code as a series of operations.~~

68. (Currently Amended) The [[A]] method according to claim 67, wherein the series of operations comprises a plurality of operations from a group of operations including ~~by which the variable is presented is made up of~~ arithmetic operations and ~~and/or~~ logical operations.~~, and wherein the series of operations is arranged, upon running of the object code, to provide the variable.~~

69. (Currently Amended) The [[A]] method according to claim 68, wherein the group of operations further includes operations which can reliably return a variable to its assigned value. ~~series of operations by which the variable is presented comprises complementary operations arranged, upon running of the object code, to provide the variable.~~

Claims 70-71 (Cancelled).

72. (Currently Amended) The [[A]] method according to claim 67, wherein the source code is written in the C^{++} programming language.

Claims 73-77 (Cancelled).

78. (Currently Amended) A method of producing storage media having a secured executable program thereon, the method comprising: ~~the steps of~~ obfuscating object code of a security program by substituting an assignment of a variable in source code of the security program with a class template defining a plurality of functions of the variable, each of the plurality of functions indexed by a key value and associated with a series of operations resulting in the assignment of the variable in a manner obfuscating such assignment, and compiling the source code using a compiler to produce the object code, wherein the compiler inserts in the object code the series of operations associated with one of the plurality of functions of the variable identified by a key value provided as a parameter of the class template; securing an executable program by associating the executable program with the [[a]] security program which is arranged to control access to the executable program;[[,]] and applying the secured executable program to a storage media.~~, and the method further comprising obfuscating the object code of the security program, wherein the object code of the security program has been obfuscated by substituting a variable in source code with a selected function of the variable, and compiling the source code to produce object code, the selected function causing the variable to be presented in the compiled object code as a series of operations.~~

79. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, wherein the executable program and the security program are associated at object code level, the security program being arranged to encrypt the executable program.

80. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, further comprising

moving blocks of the executable program out of the executable program and relocating the blocks in the security program.

81. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, wherein the security program is arranged to require the running of an authentication program.

82. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, wherein the source code of the security program involves stored arrays and templates and utilizes ~~utilises~~ pointers to navigate the arrays and templates.

83. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, wherein the source code of the security program is written in the C^{++} programming language.

84. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, wherein the storage media onto which the secured executable program is applied is an optical disc.

85. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 84, wherein the secured executable program is applied to the optical disc by laser beam encoding.

86. (Currently Amended) The [[A]] method ~~of producing storage media having a secured executable program thereon~~ according to claim 78, wherein the storage media onto which the secured executable program is applied is a memory unit ~~in, or~~ associated with at least one computer.~~, servers, computers and/or other processing means.~~

87. (Currently Amended) A storage media having an ~~a secured~~ executable program and a security program thereon, wherein ~~an executable program is secured by having a security program associated therewith,~~ the security program controls ~~being arranged to control~~ access to the executable program, ~~and wherein~~ the security program is in object code, and the object code of the security program [[which]] has been obfuscated

by substituting an assignment of a variable in source code of the security program with a class template defining a plurality of functions of the variable, each of the plurality of functions indexed by a key value and associated with a series of operations resulting in the assignment of the variable in a manner obfuscating such assignment, and compiling the source code using a compiler to produce the object code, wherein the compiler inserted in the object code the series of operations associated with one of the plurality of functions of the variable identified by a key value provided as a parameter of the class template.~~, the security program having been compiled from source code including variables, and a variable in the source code of the security program having been compiled to be presented in object code as a series of operations whereby the object code has been obfuscated.~~

88. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein the series of operations comprises a plurality of operations from a group of operations including ~~by which the variable is presented is made up of~~ arithmetic operations and ~~and/or~~ logical operations.~~, and wherein the series of operations is arranged, upon running of the object code, to provide the variable.~~

89. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 88, wherein the group of operations further includes operations which can reliably return a variable to its assigned value. ~~series of operations by which the variable is presented comprises complementary operations arranged, upon running of the object code, to provide the variable.~~

90. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein the executable program and the security program are associated at object code level, and wherein the executable program is encrypted on the storage media and the associated security program enables decryption of the executable program.

91. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein blocks from the executable program have been relocated within the security program.

92. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein the security program is arranged to require the running of an authentication program.

93. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein the storage media is an optical disc on which the executable program and the security program are encoded.

94. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 93, wherein the optical disc is one of: a CD, a CD-ROM, and a DVD.

95. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein the storage media is a memory unit ~~in, or~~ associated with at least one computer.~~, servers, computers and/or processing means and on which the executable program and the security program are stored.~~

96. (Currently Amended) The [[A]] storage media ~~having a secured executable program thereon~~ according to claim 87, wherein the executable program is one or more of: a games program; a video program; an audio program; and other software.

97. (Currently Amended) A method implemented by a first processor for providing a program for secure execution on a second processor, the method comprising: ~~of controlling a processor to run a program comprising:~~ receiving a key value; using the key value to retrieve information of a template from a library of templates indexed by key values, wherein each of the templates in the library is associated with a different encrypted set of instructions that result from compiling source code of the program and associated with a different emulator program configured to translate corresponding of the

different encrypted set of instructions into a set of obfuscated native instructions; compiling the source code to generate an encrypted set of instructions associated with the template, wherein the encrypted set of instructions is not directly executable by the second processor; and providing the encrypted set of instructions along with an emulator program associated with the template to the second processor, wherein the emulator program and the set of obfuscated native instructions are directly executable by the second processor. ~~translating instructions from the program into a reduced instruction set format to which said processor is not responsive, causing the translated instructions to be applied to a virtual processor which is responsive to the reduced instruction set format, and causing the virtual processor to run the instructions applied thereto and to apply a series of simple instructions, to which the processor is responsive, to the processor.~~

98. (Currently Amended) The [[A]] method ~~of controlling a processor to run a program~~ according to claim 97, wherein the emulator program is configured to translate the encrypted set of instructions into the associated set of obfuscated native instructions without decrypting the encrypted set of instructions. ~~further comprising encrypting the translated instructions to be applied to the virtual processor, and enabling the virtual processor to respond to the encrypted instructions without decrypting them.~~

99. (Currently Amended) The method according to claim 97, further comprising providing an executable program along with the encrypted set of instructions and the emulator program to the second processor, wherein the encrypted set of instructions control access to the executable program so that the second processor is allowed to execute the executable program. ~~A method of controlling a processor to run a program according to claim 98, further comprising utilising templates to translate and encrypt the instructions, a selected template providing a series of instructions in the reduced instruction set format for each instruction from the program, wherein the templates define a plurality of series of instructions in the reduced instruction set format for an instruction in the program, the method further comprising selecting one of said plurality of series of instructions to be the translation for said instruction.~~

100. (Currently Amended) The method according to claim 97, wherein the emulator program translates the encrypted set of instructions into the obfuscated set of native instructions by translating at least one of the encrypted set of instructions into a string of native instructions to obfuscate the native instructions. ~~A method of controlling a processor to run a program according to claim 99, wherein a different key is associated with each one of the plurality of series of instructions in the reduced instruction set format in the template, and wherein the method further comprises selecting a key which is associated with one of said plurality of series of instructions and translating the instruction in the program to the one of said plurality of series of instructions which is associated with the selected key.~~

Claims 101-105 (Cancelled).